Assignment Submission

Lanston Chen

November 4, 2023

Contents

1 Submission Details

1.1 RDS Endpoint and Table Names

RDS endpoint: `database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com`
List of table names:

- happy2015
- happy2016
- happy2017
- happy2018
- happy2019

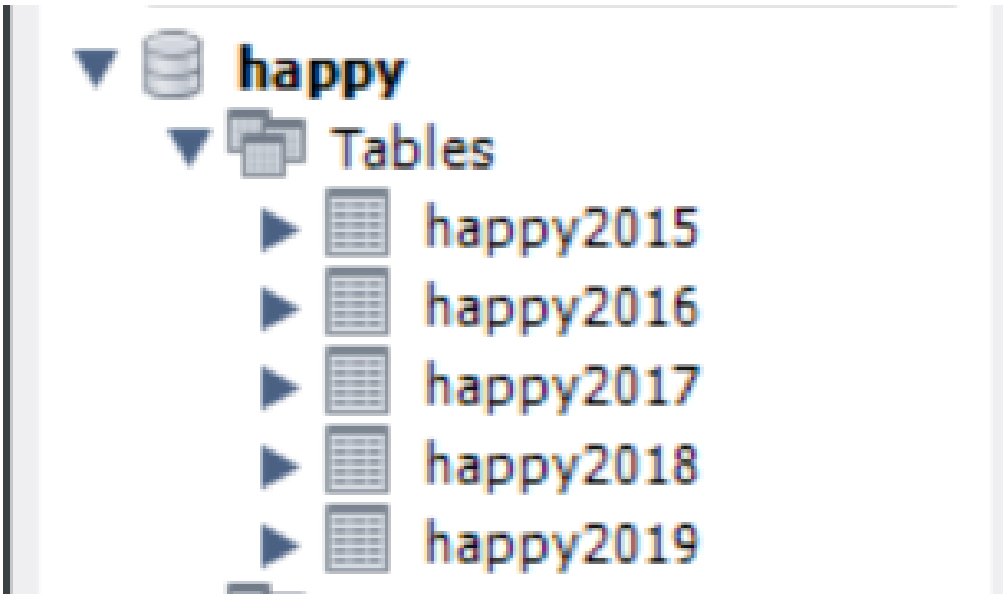

Figure 1: Table Name

1.1.1 RDS code (create table / load data/ create primary key)

```
literate
import os
import pandas as pd
from sqlalchemy import create_engine

# Database credentials
host = "database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com"
port = "3306"
user = "admin"
```

```
password = "Ct123456"
database = "happy"

# Directory containing CSV files
directory = "C:\\Users\\ctlan\\OneDrive\\desktop\\manage␣
    bigdata\\assignment\\HW\\hw5\\worldhappiness\\"

# Create a connection to the database
engine =
    create_engine(f'mysql+mysqlconnector://{user}:{password}@{host}:{port}/{database}')

# Define a dictionary for the year and its corresponding happiness score
    column name
happiness_score_columns = {
    '2015.csv': 'Happiness␣Score',
    '2016.csv': 'Happiness␣Score',
    '2017.csv': 'Happiness.Score',
    '2018.csv': 'Score',
    '2019.csv': 'Score'
}

# Process each CSV file specified in the dictionary
for filename, happiness_column in happiness_score_columns.items():
    file_path = os.path.join(directory, filename)
    table_name = str("happy") + (str(os.path.splitext(filename)[0]))  # Use
    filename without '.csv' as table name

    # Load the CSV file into a pandas DataFrame
    df = pd.read_csv(file_path)

    # Standardize column name to 'Happiness Score'
    df.rename(columns={happiness_column: 'Happiness␣Score'}, inplace=True)

    # Check if 'Country' or 'Country or region' is the column name and
    standardize
    if 'Country' in df.columns:
        country_column = 'Country'
    elif 'Country␣or␣region' in df.columns:
        country_column = 'Country␣or␣region'
    else:
        raise ValueError('Country␣column␣not␣found␣in␣the␣file␣' + filename)

    # Sort by 'Happiness Score' and 'Country', reset index to create 'id'
    column
```

```
    df.sort_values(by=['Happiness Score', country_column],
    ascending=[False, True], inplace=True)
    df.reset_index(drop=True, inplace=True)
    df.index += 1  # Make 'id' start from 1 instead of 0
    df.reset_index(inplace=True)
    df.rename(columns={'index': 'id'}, inplace=True)

    # Create or replace the table and load data
    df.to_sql(name=table_name, con=engine, if_exists='replace',
    index=False, chunksize=500)

    print(f"Data from {file_path} has been loaded into the {table_name}
    table.")

print("All specified CSV files have been processed and corresponding tables
    created.")
\end{linsting}

\subsection{S3 Bucket URI}
S3 bucket URI: \texttt{s3://lanston-lb5-happy671/671data/happy\_2018 }

\subsection{EMR Master Public DNS}
EMR master public DNS for SSH:
    \texttt{ec2-54-162-119-106.compute-1.amazonaws.com}

\subsection{EMR Hue Application Interface Link}
EMR Hue interface link:
    \href{http://ec2-54-197-65-78.compute-1.amazonaws.com:8888/hue/oozie/editor/workflow/
    Website}
{workflow=18}

\subsection{Sqoop Command Executions}
\begin{lstlisting}[language=bash]
sqoop import \
--connect
    jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.rds.amazonaws.com/happy
    \
--username admin \
--password Ct123456 \
--table happy2018 \
--target-dir /user/hadoop/lab5 \
--split-by id
```

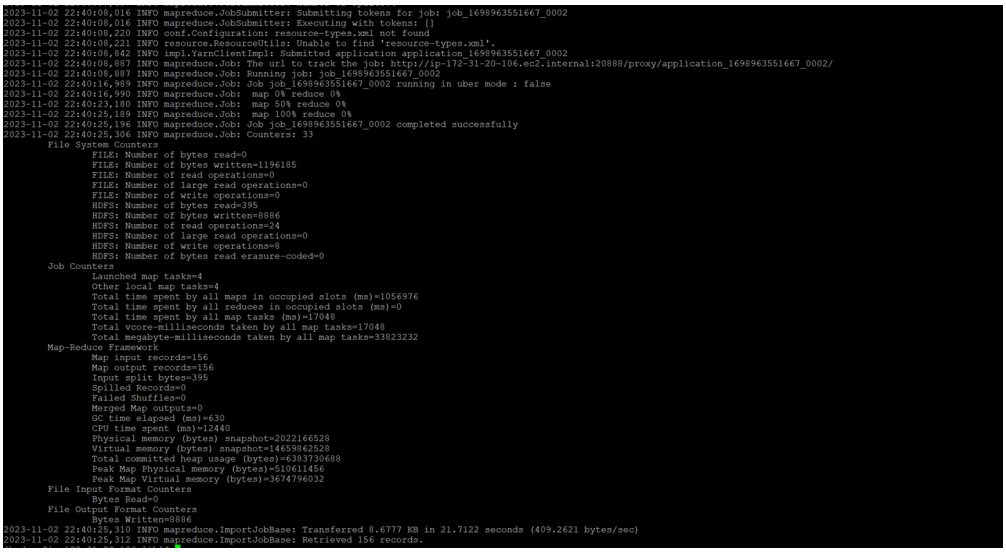

Figure 2: Sqoop Command Executions

1.2 List of Files in S3 Bucket

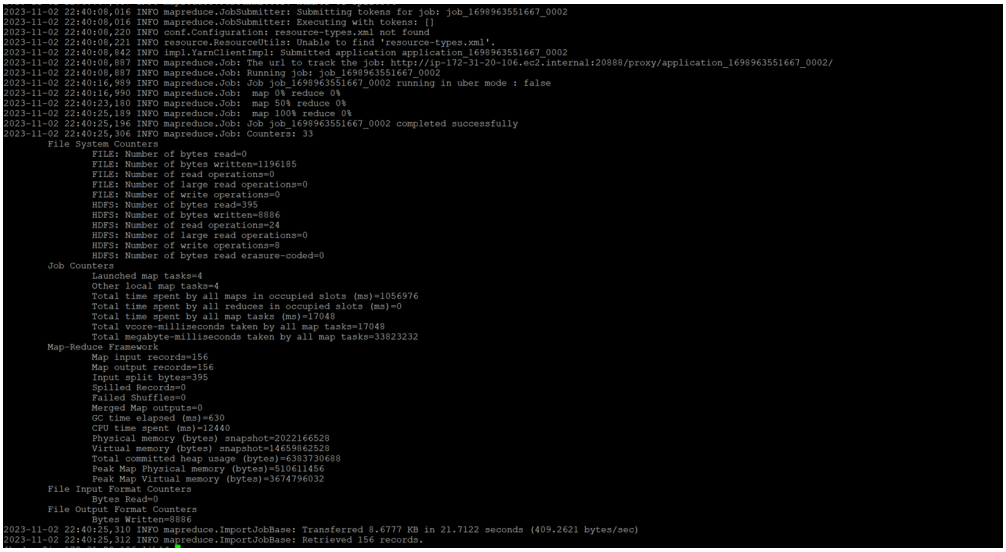

Figure 3: S3 Bucket

1.3 List of Files in Hadoop Folder

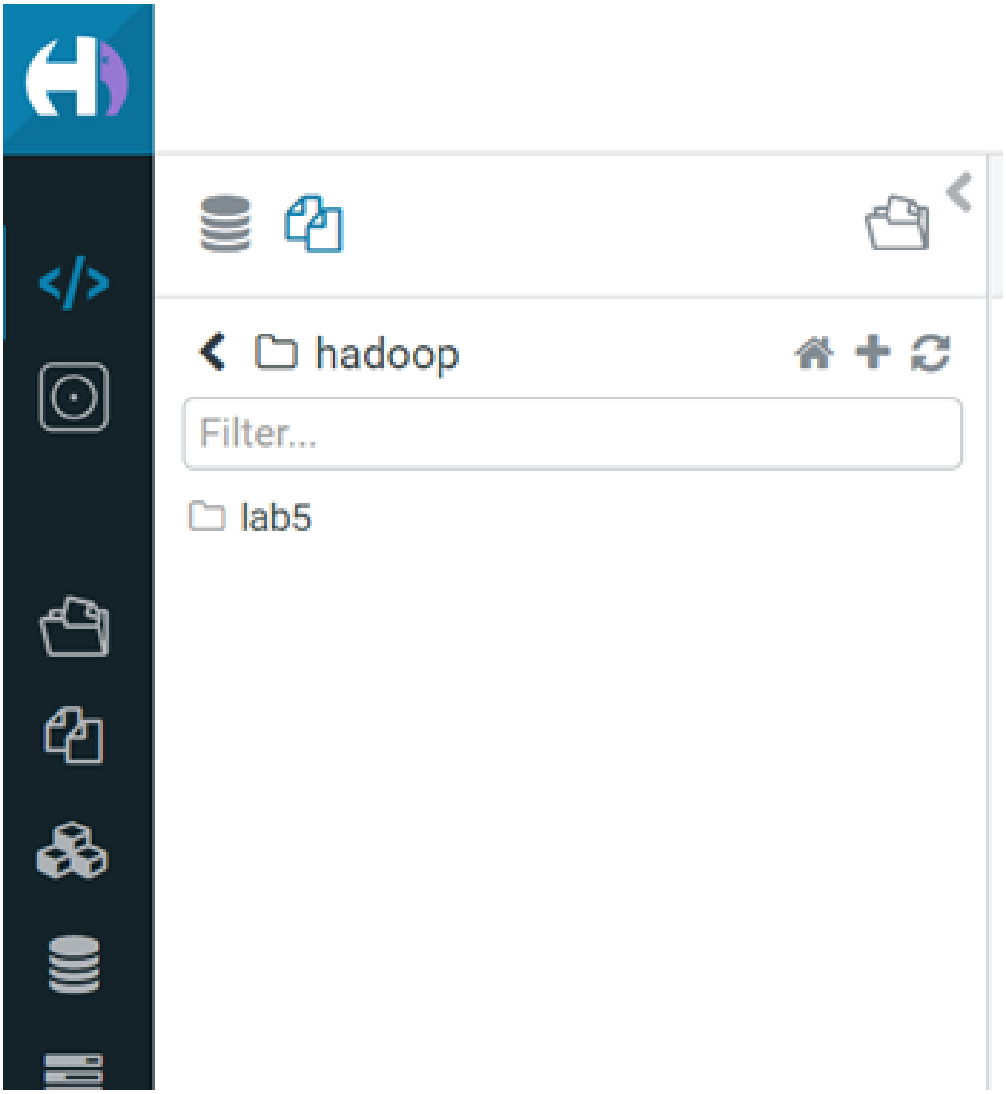

Figure 4: Hadoop Folder

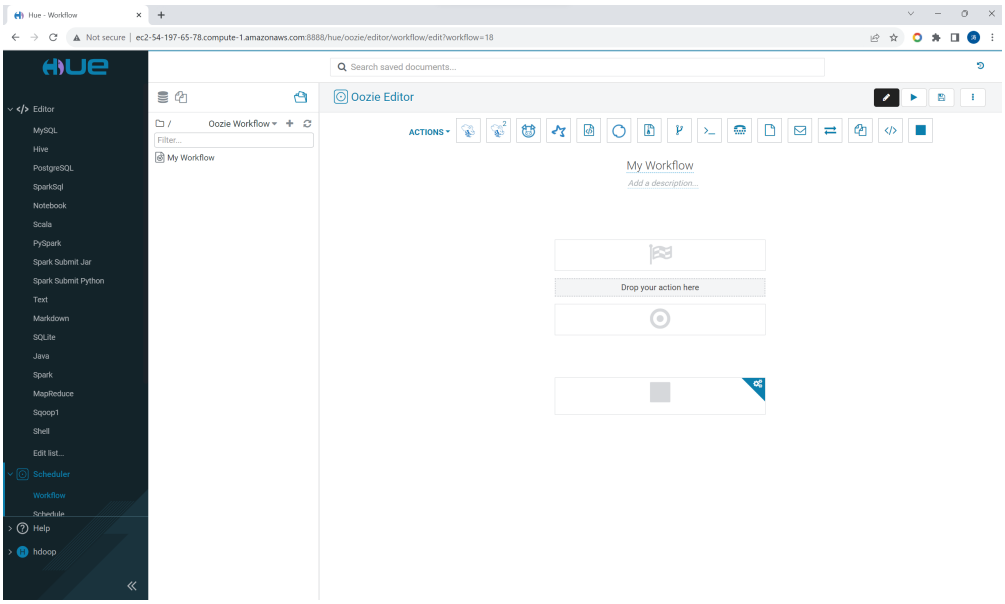

Figure 5: Hadoop Interface

1.4 Oozie Workflow Completion or XML File

```
<!-- <workflow-app name="Lab 5 Workflow" xmlns="uri:oozie:workflow:0.5">
    <start to="sqoop-71ff"/>
    <kill name="Kill">
        <message>Action failed, error message[\$\{wf:errorMessage(wf:lastErrorNode())\}]
    </kill>
    <action name="sqoop-71ff">
        <sqoop xmlns="uri:oozie:sqoop-action:0.2">
            <job-tracker>\$\{jobTracker\}</job-tracker>
            <name-node>\$\{nameNode\}</name-node>
            <command>import --connect jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.
        </sqoop>
        <ok to="sqoop-7711"/>
        <error to="Kill"/>
    </action>
    <action name="sqoop-7711">
        <sqoop xmlns="uri:oozie:sqoop-action:0.2">
            <job-tracker>\$\{jobTracker\}</job-tracker>
            <name-node>\$\{nameNode\}</name-node>
            <command>import --connect jdbc:mysql://database-lab5.c3f4g2yxdyve.us-east-1.
        </sqoop>
        <ok to="email-98f8"/>
        <error to="Kill"/>
    </action>
    <action name="email-98f8">
        <email xmlns="uri:oozie:email-action:0.2">
```

```
            <to>tche368@emory.edu</to>
            <subject>Workflow Notification: Import Complete</subject>
            <body>The Sqoop import actions have completed successfully.</body>
            <content_type>text/plain</content_type>
        </email>
        <ok to="End"/>
        <error to="Kill"/>
    </action>
    <end name="End"/>
</workflow-app>
 -->
```